



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/892,633	06/28/2001	Randal F. Templeton	219.40067X00 (ATSK)	4474
7590	11/14/2007		EXAMINER	
Kenyon & Kenyon 1500 K Street, N.W. Suite 700 Washington, DC 20005-1257			TRAN, QUOC A	
			ART UNIT	PAPER NUMBER
			2176	
			MAIL DATE	DELIVERY MODE
			11/14/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	09/892,633	TEMPLETON ET AL.
	Examiner	Art Unit
	Tran A. Quoc	2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 30 August 2007.

2a) This action is FINAL. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-18 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-18 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.	5) <input type="checkbox"/> Notice of Informal Patent Application
	6) <input type="checkbox"/> Other: _____.

DETAILED ACTION

This is a **Non-Final Rejection** in response to Remarks filed 08/30/2007. Claims 1-18 are pending. Claims 1, 3-4, 7-8, 10-11, 13, and 16 are previously presented. Claims 2, 5-6, 9, 12, 14-15, and 17-18 are originally presented. Effective filing date is 06/28/2001.

Claims Rejections – 35 U.S.C. 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-6, and 16-18 are directed to non-statutory subject matter.

As evidence, claim 1 is directed to “*A system for generating and communicating to web pages, comprising:*” This claimed subject matter lacks a practical application of a judicial exception (law of nature, abstract idea, naturally occurring article/phenomenon) since it fails to produce a useful, concrete and tangible result. Specifically, the claimed subject matter does not produce a tangible result because the claimed subject matter fails to produce a result that is limited to having real world value rather than a result that may be interpreted to be abstract in nature as, for example, a thought, a computation, or manipulated data. More specifically, the claimed subject matter provides for “*A system for generating and communicating to web pages, comprising:*” The phase “*system*” is not clearly recites hardware/software, however the Applicant specification discloses, “*system, method and computer program is provided for generating web pages and communicating between web pages- See Applicant specification Para 1, 4, and 23.*

This produced result remains in the abstract and, thus, fails to achieve the required status of having real world value.

In addition, claim 16 is directed to “*A computer program for communicating between web pages, comprising:*” This claimed subject matter lacks a practical application of a judicial exception (law of nature, abstract idea, naturally occurring article/phenomenon) since it fails to produce a useful, concrete and tangible result. Specifically, the claimed subject matter does not produce; a tangible result because the claimed subject matter fails to produce a result that is limited to having real world value rather than a result that may be interpreted to be abstract in nature as, for example, a thought, a computation, or manipulated data. More specifically, the claimed subject matter provides for “*A computer program for communicating between web pages, comprising:*” The phase “*computer program*” is clearly recites software, thus this produced result remains in the abstract and, thus, fails to achieve the required status of having real world value.

In the interest of compact prosecution, the application is further examined against the prior art, as stated below, upon the assumption that the applicants may overcome the above stated rejections under 35 U.S.C. 101.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-18, are rejected under 35 U.S.C. 103(a) as being unpatentable over O'Rourke et al. US007117436B1-filed 04/31/2000 (hereinafter O'Rourke), in view of Alexander US20040205528A1- CON of 10/834,595 filed 02/15/2000 (hereinafter Alexander).

Independent claim 1, O'Rourke teaches:

A system for generating and communicating to web pages,
(See O'Rourke at Column 2, Lines 60-65, discloses a system for generating and communicating to web pages.)
comprising: a console engine to receive requests for web pages and messages to be sent to web pages;
(See O'Rourke at Column 3, Line 55 → Column 4, Line 20, discloses a controller program makes substitution calls to the HTT (Hypertext Template) engine specifying markers and dynamic content to be placed at the marker location. The HTT engine responds to substitution calls from the controller program output structure to determine the output order of the display regions and returned to the Web browser.

In the broadest reasonable interpretation in light of the Applicant's Specs- Page 6 Para 14, "*The console engine 220 would assemble the web page desired according to the HTML/XML template ... transmit the web page to the web server 21 for display by the web browser 200.*" the Examiner reads the claimed a *console engine* as equivalent to a controller program and the claimed *messages to be sent to web pages* as equivalent to output structure to determine the output order of the display regions and returned to the Web browser (i.e. browser applications for viewing Web pages on client computer systems, and Web site is actually a collection of individually downloadable Web pages (See O'Rourke at Column 1, Lines 15-35).)

and an XML repository connected to the console engine having a plurality of parts of web pages and a plurality of HTML/XML templates,

(See O'Rourke at Fig.1- 2 and Column 6, Lines 15-50, discloses a detail block diagram showing the system 10 for dynamically generating Web content of FIG. 1. The server 11 consists of three functional modules: template manager 21, HTT engine 22, and database engine 23. The template manager 21 maintains an HTT template repository 32 within the database 26. Each HTT template is a modified Web page initially written as an interpretable script in a tag-delimited page description language, such as HTML or XML. Markers are embedded into the script at locations where dynamic content will appear. The template manager 21 uploads the HTT templates into the HTT template repository 32.)

said retrieved application handler being registered to said extracted template and said application handler to modify said template and to generate a part of said requested web page and incorporate that part into the template to form the web page, and said console engine is to retrieve at least one application handler,

(See O'Rourke at Fig.1- 2 and Column 6, Lines 15-50, discloses a detail block diagram showing the system 10 for dynamically generating Web content of FIG. 1. The server 11 consists of three functional modules: template manager 21, HTT engine 22, and database engine 23. Each HTT template is a modified Web page initially written as an interpretable script in a tag-delimited page description language, such as HTML or XML. Markers are embedded into the script at locations where dynamic content will appear. The template manager 21 uploads the HTT templates into the HTT template repository 32.

In the broadest reasonable interpretation in light of the Applicant 's Specs- Page 6 Para 15, "*the console engine 220 checks the XML repository 230 for application handlers which are registered to modify the specific template,*" the Examiner reads the claimed *application handler* as equivalent to template manager item 21, which is handling a controller program makes substitution calls to the HTT (Hypertext Template) engine specifying markers and dynamic content to be placed at the marker location. The HTT engine responds to substitution calls from the controller program output structure to determine the output order of the display regions and returned to the Web browser as taught by O'Rourke.

In addition, O'Rourke does not expressly teach, but Alexander teaches:

**wherein the console engine is to extract a template for a web page
from one of said requests,**

(See Alexander at Fig. 5 and Para 73 discloses content management framework object model item 60, includes CMMForms, which contains CMFTemplates that defines the XML template. The content management framework retrieves XML template upon a request from a client, then combined with metadata retrieved from the SQL server database 57 and stored Web templates and style sheets 58. It is noted, the Web templates are written as proprietary scripts, such as Microsoft Active Server Pages, and the style sheets are written in the Extensible Stylesheet Language (XSL). The resultant Web page 59 is served to the requesting client. Alternatively, the XML document 56 can be combined with the stored web templates and style sheets, and sends directly by the web server.)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified O'Rourke XML repository connected to the console engine having a plurality of parts of web pages and a plurality of HTML/XML templates, to include a means of utilizing the console engine is to extract a template for a web page from one of said requests as taught by Alexander, in order to edit HTML/XML scripts at runtime using conventional Web publishing tools directly manipulated by a page artist (See O'Rourke at Column 2, Lines 5-20, and at Column 6, Lines 20-25).

Independent Claim 7:

The rejection of claim 1 is fully incorporated;

In addition O'Rourke teaches:

Executing the at least one application handler to generate a plurality of parts for the web page; combining the plurality of parts for the web page with the template to form the web page; and transmitting the web page to the web browser for display.

(See O'Rourke at Column 3, Line 55 → Column 4, Line 20, discloses the controller program indicates that it is ready for the Web page to be displayed. a controller program makes substitution calls to the HTT (Hypertext Template) engine specifying markers and dynamic content to be placed at the marker location. The HTT engine responds to substitution calls from the controller program output structure to determine the output order of the display regions and returned to the Web browser.

In the broadest reasonable interpretation in light of the Applicant 's Specs- Page 6 Para 15, "*the console engine 220 checks the XML repository 230 for application handlers which are registered to modify the specific template,*" the Examiner reads the claimed *application handler* as equivalent to template manager item 21, which is handling a controller program makes substitution calls to the HTT (Hypertext Template) engine specifying markers and dynamic content to be placed at the marker location. The HTT engine responds to substitution calls from the controller program output structure to determine the output order of the display regions and returned to the Web browser as taught by O'Rourke.

Independent Claim 10:

is directed to a computer program to perform the method recited in Claim 7 and is similarly rejected along the same rationale (See O'Rourke at Column 3, Line 45, i.e. Controller program.)

Independent claim 13, O'Rourke teaches:

**A method of communicating between web pages, comprising:
receiving an incoming XML data element from a source web page;
(See O'Rourke at Column 2, Lines 60-65, discloses a system for generating and
communicating to web pages.)
and displaying the modified XML data element using a web browser, said
modified XML data element including a template for the destination web
page.**

(See O'Rourke at Column 3, Line 55 → Column 4, Line 20, discloses the controller program indicates that it is ready for the Web page to be displayed. a controller program makes substitution calls to the HTT (Hypertext Template) engine specifying markers and dynamic content to be placed at the marker location. The HTT engine responds to substitution calls from the controller program output structure to determine the output order of the display regions and returned to the Web browser.

Also see O'Rourke at Fig.1- 2 and Column 6, Lines 15-50, discloses a detail block diagram showing the system 10 for dynamically generating Web content of FIG. 1. The server 11 consists of three functional modules: template manager 21, HTT engine

22, and database engine 23. The template manager 21 maintains an HTT template repository 32 within the database 26. Each HTT template is a modified Web page initially written as an interpretable script in a tag-delimited page description language, such as HTML or XML. Markers are embedded into the script at locations where dynamic content will appear. The template manager 21 uploads the HTT templates into the HTT template repository 32.)

In addition O'Rourke does not expressly teach, but Alexander teaches:

parsing the incoming XML data element based on delimiters to determine the source web page, a destination web page, and data to be received by the destination web page;

(See Alexander at Para(s) 28-36, discloses incoming XML data element based on tag-delimited language such as HTML/XML in server/client environment utilized known technology such as JavaBean, Active Server Page (ASP), and the like. The server include two applications, a web server and a content management framework (i.e. source web page) operates at a Metadata level, enable a user to manage the arrangement, composition, and display attributes of Web page content as maintained in a data store and database as well as the uses of share module, which enables data to be exchanged across systems or organization boundaries.)

creating a pretoken from the data in the incoming XML data element; concatenating the pretoken to a token to form a modified XML data element;

(See Alexander at Fig.9 and Para(s) 76-77, discloses retrieving the XML content containing the content data for the data entry form (block 189). The XML content can come from several sources. The XML content can be stored as session state (i.e. *pretoken*) with the host environment. The retrieved XML content is then updated with the new data values received as input parameters of the HTTP "POST" request (block 190); there are three form-updating options. First, if either the "+" or "-" button is pressed (block 191); XML nodes are added or removed from the XML content (block 194). The controls can be hierarchically structured with each successive generation of controls contained in the parent control class (i.e. *pretoken* to a *token*).)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified O'Rourke XML repository connected to the console engine having a plurality of parts of web pages and a plurality of HTML/XML templates, to include a means of creating a pretoken from the data in the incoming XML data element; concatenating the pretoken to a token to form a modified XML data element, and creating a pretoken from the data in the incoming XML data element; concatenating the pretoken to a token to form a modified XML data element as taught by Alexander, in order to edit HTML/XML scripts at runtime using conventional Web publishing tools directly manipulated by a page artist (See O'Rourke at Column 2, Lines 5-20, and at Column 6, Lines 20-25).

Independent Claim 16:

is directed to a computer program to perform the method recited in Claim 13 and is similarly rejected along the same rationale (See O'Rourke at Column 3, Line 45, i.e., Controller program.)

Claim 2, O'Rourke teaches:

a web browser to request the web page from the console engine and display the web page.

(See O'Rourke at Column 3, Line 55 → Column 4, Line 20, discloses a controller program makes substitution calls to the HTT (Hypertext Template) engine specifying markers and dynamic content to be placed at the marker location. The HTT engine responds to substitution calls from the controller program output structure to determine the output order of the display regions and returned to the Web browser.)

Claim 3, O'Rourke teaches:

an XML repository to contain the plurality of parts of web pages, the plurality of HTML/XML templates and a plurality of said application handlers.

(See O'Rourke at Fig.1- 2 and Column 6, Lines 15-50, discloses a detail block diagram showing the system 10 for dynamically generating Web content of FIG. 1. The server 11 consists of three functional modules: template manager 21, HTT engine 22, and database engine 23. The template manager 21 maintains an HTT template repository

32 within the database 26. Each HTT template is a modified Web page initially written as an interpretable script in a tag-delimited page description language, such as HTML or XML. Markers are embedded into the script at locations where dynamic content will appear. The template manager 21 uploads the HTT templates into the HTT template repository 32.)

Claim 4, O'Rourke teaches:

a console API to transmit the web page to a web browser.

(See O'Rourke at Column 3, Line 55 → Column 4, Line 20, discloses a controller program makes substitution calls to the HTT (Hypertext Template) engine specifying markers and dynamic content to be placed at the marker location. The HTT engine responds to substitution calls from the controller program output structure to determine the output order of the display regions and returned to the Web browser.)

Claims 5-6,

the rejection of claim 13 is fully incorporated and is similarly rejected along the same rationale.

Claim 8,

the rejection of claim 4 is fully incorporated and is similarly rejected along the same rationale.

Claim 9, O'Rourke teaches:

converting the template after combining the plurality of parts for the web page with the template to form the web page into HTML so as to be displayed by the browser.

(See O'Rourke at Column 3, Line 55 → Column 4, Line 20, discloses a controller program makes substitution calls to the HTT (Hypertext Template) engine specifying markers and dynamic content to be placed at the marker location. The HTT engine responds to substitution calls from the controller program output structure to determine the output order of the display regions and returned to the Web browser.

Also see O'Rourke at Column 4, Lines 20-35, discloses the Web page for dynamic content insertion, The Web page script is served into an output buffer with the dynamic content included therein.

Also see O'Rourke at Column 2, lines 25-30, describes the HTML code most often cannot be directly manipulated by a page artist.)

Claim 11,

the rejection of claim 4 is fully incorporated and is similarly rejected along the same rationale.

Claim 12,

the rejection of claim 9 is fully incorporated and is similarly rejected along the same rationale.

Claim 14, O'Rourke teaches:

wherein incoming XML data element is a portion of a web page in which that data to be displayed is changing, and said token is an existing web page.

(See O'Rourke at Fig.1- 2 and Column 6, Lines 15-50, discloses a detail block diagram showing the system 10 for dynamically generating Web content of FIG. 1. The server 11 consists of three functional modules: template manager 21, HTT engine 22, and database engine 23. The template manager 21 maintains an HTT template repository 32 within the database 26. Each HTT template is a modified Web page initially written as an interpretable script in a tag-delimited page description language, such as HTML or XML. Markers are embedded into the script at locations where dynamic content will appear. The template manager 21 uploads the HTT templates into the HTT template repository 32.)

In addition, O'Rourke does not expressly teach, but Alexander teaches:

and said token is an existing web page.

(See Alexander at the Abstract, discloses dynamically generating Web content using a parse tree is described. A template describing a dynamically generated Web page is built. The template includes a script written in a tag-delimited page description language. One or more markers included within the template each indicate a relative location within the Web page for dynamic content insertion.

Also See Alexander at Fig.9 and Para(s) 76-77, discloses retrieving the XML content containing the content data nodes, which are added or removed from the XML

content (block 194). The controls can be hierarchically structured with each successive generation of controls contained in the parent control class (i.e. *pretoken/token*).)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified O'Rourke XML repository connected to the console engine having a plurality of parts of web pages and a plurality of HTML/XML templates, to include a means of creating a pretoken/token from the data in the incoming XML data element as taught by Alexander, in order to edit HTML/XML scripts at runtime using conventional Web publishing tools directly manipulated by a page artist (See O'Rourke at Column 2, Lines 5-20, and at Column 6, Lines 20-25).

Claim 15, O'Rourke teaches:

wherein said modified XML data element is the web page to be displayed.

(See O'Rourke at Fig.1- 2 and Column 6, Lines 15-50, discloses a detail block diagram showing the system 10 for dynamically generating Web content, consists of three functional modules: template manager 21, HTT engine 22, and database engine 23. The template manager 21 maintains an HTT template repository 32 within the database 26. Each HTT template is a modified Web page initially written as an interpretable script in a tag-delimited page description language, such as HTML or XML.

Also see O'Rourke at Column 3, Line 55 → Column 4, Line 20, discloses a controller program makes substitution calls to the HTT (Hypertext Template) engine specifying markers and dynamic content to be placed at the marker location. The HTT

engine responds to substitution calls from the controller program output structure to determine the output order of the display regions and returned to the Web browser.)

Claims 17-18:

are directed to a computer program to perform the method recited in Claim 14-15 and are similarly rejected along the same rationale (See O'Rourke at Column 3, Line 45, i.e. Controller program.)

It is noted that any citations to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. See, MPEP 2123.

Response to Arguments

The Remarks filed on 08/30/2007 has been fully considered but are moot but in view of the new ground(s) of rejection. This office action is a Non-Final Rejection in order to give the applicant sufficient opportunity to response to the new line of rejection (see rejection for details).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is 571-272-8664. The examiner can normally be reached on Monday through Friday from 9 AM to 5 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached on 571-272-4137. The fax phone number for the organization where this application or proceeding is assigned is (571)-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Quoc A. Tran/
Patent Examiner
Art Unit 2176
11/10/2007

/Doug Hutton/
Doug Hutton
Supervisory Primary Examiner
Technology Center 2100